RESEARCH ARTICLE OPEN ACCESS

# An LSB Method Of Image Steganographic Techniques

Lalit Kumar Jain[1], Jyotiprakash Patra[2], Himanshu Kumar Gawhade3
Dept. Of CSE SSIPMT RAIPUR, CG, INDIA
Associate Professor, Dept. Of CSE SSIPMT RAIPUR, CG, INDIA
Dept of CSE SSIPMT RAIPUR, CG, INDIA

***Abstract***
The art of information hiding has received much attention in the recent years as security of information has become a big concern in this internet era. As sharing of sensitive information via a common communication channel has become inevitable. Steganography means hiding a secret message (the embedded message) within a larger one (source cover) in such a way that an observer cannot detect the presence of contents of the hidden message [1]. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet [2]. This paper intends to give an overview of image Steganography, its uses and techniques. It also attempts to identify the requirements of a good Steganography algorithm and briefly reflects on which Steganography techniques are more suitable for which applications.
***Keywords***— Steganography, LSB method, Stego-object, Spatial domain, ARGB.

## I. INTRODUCTION

Steganography is the art and science of hiding information has gained much attention in the recent years as security of information has become a big concern in this internet era. Steganography derives from the Greek word steganos, meaning covered or secret, and graphia meaning writing. The main purpose of Steganography, which means 'writing in hiding' is to hide data in a cover media so that others will not be able to notice it. While cryptography is about protecting the content of messages, steganography is about concealing their very existence. The general model of data hiding can be described as follows. The embedded data is the message that one wishes to send secretly. It is usually hidden in an innocuous message referred to as a cover-text or cover-image or cover-audio as appropriate, producing the stego-text or other stego-object [3]. The applications of information hiding systems mainly range over a broad area from military, intelligence agencies, online elections, internet banking, medical-imaging and so on. These variety of applications make steganography a hot topic for study. The cover medium is usually chosen keeping in mind the type and the size of the secret message and many different carrier file formats can be used. In the current situation digital images are the most popular carrier/cover files that can be used to transmit secret information.

The basic model for Steganography is shown on Fig 1. It shows the basic process involved in Steganography which consists of Carrier, Message and Key. Carrier is also known as cover-object, in which message is embedded and serves to hide the presence of the message. The data can be any type of data (plain text, cipher text or other image) that the sender wishes to remain confidential. Key is known as stego-key, which ensures that only recipient who knows the key, corresponding decoding key will be able to recover the message from a cover-object. The cover-object with the object secretly embedded message is then called the stego-object [4].

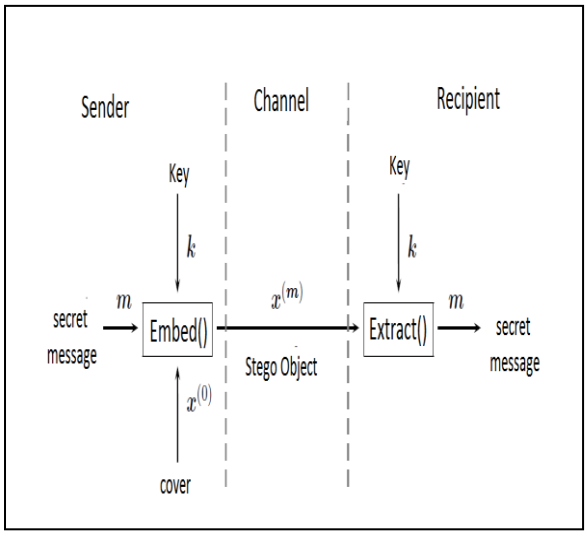


Fig 1: BASIC STEGANOGRAPHY MODEL [5]

## II. PRINCIPLE OF STEGANOGRAHY

The secret message is embedded inside the cover object in encrypted format by using a hiding algorithm and it sent to a receiver over a network. The receiver then decrypted the message by applying the reverse process on the cover data and reveals the secret data.

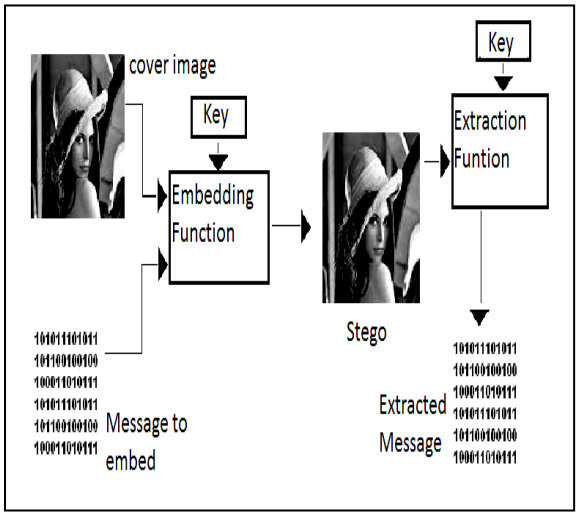


Fig. 2: THE PRINCIPLE OF STEGANOGRAPHY

Fig.2 shows the principle of Steganography. Steganography algorithm, tries to preserve the perceptive properties of the original image. A suitable image, called as cover/ carrier, is chosen. The secret message is then embedded into the cover using the Steganography algorithm, in a way that does not change the original image in a human noticeable way. The result is new image, the stego-image, which is not looks different than original image [6].

## III. PROJECT METHODOLOGY

### *A. Format Of Image Data*

Images are constructed using tiny dots named as pixels. Each pixel has got its own attributes for displaying color and transparency. There are several systems available for representing image pixels. The most common system for representing color is the ARGB system, which stores pixels data in the form of red, green, blues & alpha (transparency). In this project we have used ARGB system for storing and manipulating pixel data.

Under ARGB system, first 8 bits (0 to 7) of the pixel belong to alpha value or the transparency value. The second 8 bits (8 to 15) represents red color, third 8 bits (16 to 24) represents green color and last 8bits (24 to 31) represents blue color.

Now we understand that the maximum value for each parameter of ARGB system is 2^8, i.e. 256. If a change is made to the value at the least significant bit, i.e. the bit location 0 for alpha value, 8 for red value, 16 for green value and 24 for blue value, the impact is likely to be 0.39%. Since the change is very low, we might use the LSB of any or all the four ARGB bytes for storing information we wish to transmit incognito [7].

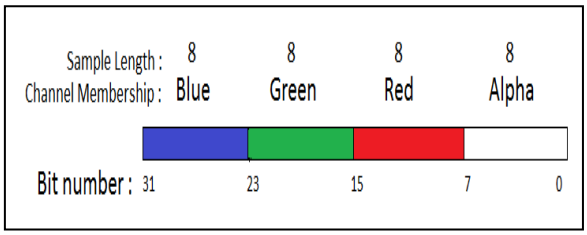


Fig 3: ARGB SYSTEM [7]

### *B. Strategy For Storing Message In An Image*

In our proposed system we have used only the least significant bit of the alpha part of a pixel, So that it does not modify any color value. Before embedding the length of the message should be written into the image. This will exclude the appearance of junk value in the decoded message.

After extracting bit number 0 from the first 32 pixels, the bits should be neatly arranged inside an integer variable to know the length of message embedded into the image. An image with 1 mega pixels might be able to store a message containing a maximum of 1,24,996 characters ((1,000,000-32)/8=1,24,996). Maximum size of message that could be embedded in an image at the rate of 1 bot from each pixel can be calculated using the relation n= (p-32)/8 (here n is the maximum length of message and p is the number of pixels [7].

### *C. Image Format Suitable For Embedding Message*

Most of the image formats stores image data in some form of compression. The compression algorithms used for image data can be divided into two broad categories:

i) Lossy Compression Algorithms (JPG,GIF)
ii) Loss-less Compression Algorithms (PNG,BMP,DEB)

Lossy compression is not suitable for steganography transmission of message, since the pixel value may be modified by the algorithm after we embed the message.

In the proposed system we have used loss-less compression algorithm for storing steganography message. Loss-less compression algorithm compresses the image, but does not make any changes to the pixel values of the original image. Our proposed system saves image in lossless compression format of image after embedding (i.e. PNG) [7].

## IV. ALGORITHM USED IN PROJECT

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain. Image also known as spatial domain techniques embed messages in the intensity of the pixels directly, while for transform also known as frequency domain, images are first

transformed and then the message is embedded in the image [8].

### *A. Least Significant Bit*

Least significant bit (LSB) Replacement is a common, simple approach to embedding information in a cover image. The least significant bit (8 bit) of some or all of the bytes inside an image is replaced with a bit of the hidden message. When using a 32-bit image, a bit of each of the red, green blue & alpha parts of pixel can be used, since they are each represented by a byte.

That is one can store 4 bits in each pixel. The image of 800 X 600pixel, can thus store a total amount of 1,920,000 bits of embedded data.

For example, 4 pixels grid for of a 32-bit image can be as follows:

(00101101 00011100 11011101 10101010)
(10100111 11000101 00001101 01010101)
(11010010 10101101 01100011 01100110)

When the number 500, which binary representation is 11110100, is embedded into the least significant bits of this part of the image.

The resulting grid is as follows:

(00101101 0001110**1** 11011101 1010101**1**)
(1010011**0** 11000101 0000110**0** 0101010**0**)
(11010010 10101101 01100011 01100110)

Above the number was embedded into the first 8 bytes of the grid, from these only the 5 underlined bits needed to be changed according to the message which is embedded. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are possible intensities of each primary color is 256, by changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be identify by the human eye, thus the message is successfully hidden in image [8].

## V. RESULTS

### *A. Interface for embedding message*

The user interface for embedding message is shown in fig 4.

After typing the message and opening the target image for embedding the message, pressing the button named *Embed* writes the message into the image. Pressing *Save* button offers to save the image in either PNG format or BMP format, because these two formats offer loss-less compression.

The embed Message method extracts bit values from the given message and embeds the value at specified location within the given pixel. This method uses embed Integer and embed Byte methods to write one integer or byte value into the image.

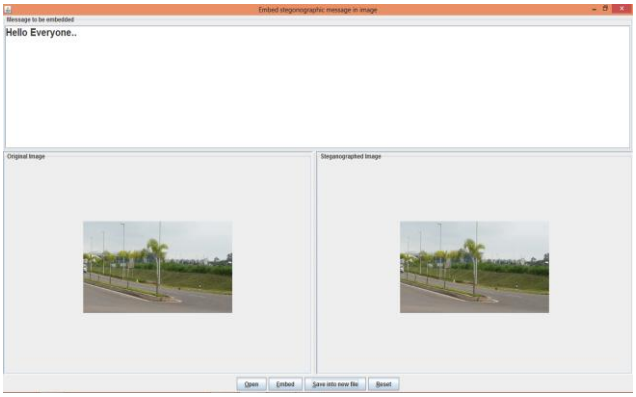


Fig 4:

### *B. Interface for decoding message*

The user interface for extracting message from an image is shown in fig 5. The image previously saved with an embedded message is opened. The message extracted from the image is shown below.

The most important method for extraction of message is decode Message, which uses extract Integer and extract Byte methods to bring out the message. The first thirty two bits of the extracted message constitute one integer representing the length of message. After knowing the length, extractByte method is used to reconstruct the byte values of the message.

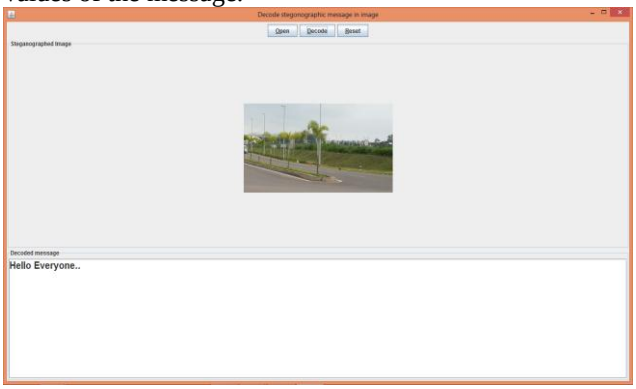


Fig 5:

## VI. CONCLUSION & FUTURE SCOPE

This paper gave an example of image steganography techniques. Although only some of the main image steganography techniques were discussed in this paper, one can see that there exists a large selection of approaches to hiding information in images. All the major file formats have different methods of hiding messages, with different strong and weak points respectively.

## References

[1] Eric Cole, Ronald D. Krutz, Consulting Editor (2003), "*Hiding in Plain Sight, Steganography and the Art of Covert Communication*", Wiley Publishing, Inc.

[2] T. Morkel, J.H.P. Eloff, M.S. Olivier, "*AN OVERVIEW OF IMAGE STEGANOGRAPHY*", *ICSA*, University of Pretoria, 0002, Pretoria, South Africa.

[3] Stefan Katzenbeiser & Fabien A.P. Petitcolas (1999), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Computer Security series, Boston, London.

[4] Jagvinder Kaur and Sanjeev Kumar, "*Study and Analysis of Various Image Steganography Techniques*", *IJCST* Vol. 2, Issue 3, September 2011.

[5] Neil F. Johnson, Zoran, Jajodia, Sushil, "*Information Hiding:Steganography And Watermarking Attacks And Countermeasures*", 1st ed., Springer US, 2001.

[6] Mr. Falesh M. Shelke, Miss. Ashwini A. Dongre, Mr. Pravin D. Soni, "*Comparision of Different Techniques for Steganography in Images*", *IJAIEM*, Vol. 3, Issue 2, Febrary 2014.

[7] Febrary 28th 2013, By Anish S, "*Embedding Messages in Digital Images*",. Available: http://developeriq.in.

[8] Johnson, N.F and Jajodia S., "*Exploring Steganography: Seeing the Unseen*", Computer Journal, February 2008.

[9] Mehdi Hussain and Mureed Hussain, "*A Survey of Image Steganography Techniques*", *IJAST*, Vol. 54, May, 2013.

[10] C. P. Sumathi, T. Santanam and G. Uma maheswari, "*A Study of Various Steganographic Techniques Used for Information Hiding*", IJCSES, Vol.4, No.6, December 2013.

[11] Jaswinder Kaur, Inderjeet & Manoj Duhan, (2009) "*A Comparative Analysis of Steganography Techniques*", International Journal of Information Technology and Knowledge Management, Vol. 2, No. 1, PP 191-194.